Интернет програмирање предавање 03

Проф. др Мирослав Лутовац

mlutovac@viser.edu.rs

Programski jezik JavaScript

- **Da prevaziđe nedostatak dinamičke obrade unetih podataka od strane korisnika**
- problem je česta klijent-server komunikacija
- unese podataka u neodgovarajućem formatu (datum rođenja, imejl adresa, ...)
- **Umesto da server proverava tačnost**, odmah na strani korisnika se prijavljuje greška

JavaScript - šta radi

Na klijentskoj strani

- Da se **formatiraju podaci**
- Da se **obrađuju podaci**
- Da se **dinamički izvršava** stranica

JavaScript - osobine

- **objektno bazirani**
- **platformski neutralan**
- višekorisnički jezik
- programeru omogućava **funkcionalnost na klijentskoj strani**
  - Primer – da klijent dobije zahtev za korekciju odmah kada ima pogrešan unos

JavaScript - objektno bazirani

- objektno bazirani – **nisu realizovani svi koncepti objektno orijentisanih jezika**
- **limitiran rad sa nasleđivanjem**, važenjem i funkcionalnošću samih objekata
- postoje hijerarhija ugrađenih objekata i oni se mogu koristiti, sa već definisanim metodama i osobinama
- dobijeno je na **jednostavnosti** samog jezika, a pomoću ugrađenih objekata **nije izgubljena potrebna funkcionalnost**

JavaScript - neutralan jezik

- **platformski neutralan** jezik (kao HTML)
- ako je kod pisan po standardu, trebalo da se **izvršava** u okviru pregledača klijenta, **bez obzira koja je vrsta hardvera** u računaru ili koje je softversko okruženje u pitanju
- **veličina programa je mala**
- može da se izvršava i **na računarima sa lošijim perfomansama**

JavaScript – modularno programiranje

- omogućava **modularno programiranje – kreiraju se svoji sopstveni objekti**
- definišu se opšte funkcije koje će realizovati uobičajene zadatke i čuvati i izvršavati kod **pomoću posebnih dokumenata**
- ekstenzija **.js**
- da se **funkcija više puta izvrši tokom aplikacije**

JavaScript – spoljašnji dokument

- Ako se realizuje spoljašnji dokument koji će sadržati ovu funkciju, koja kao argument prihvata uneti tekst, na različitim mestima upotrebe je dovoljno **samo pozvati realizovanu funkciju**
- **Ako se nešto menja, sve promene se izvršavaju samo na jednom mestu**, u eksternom fajlu (JavaScript dokumentu)

JavaScript - integrisanost

- **integrisan sa HTML**-om
- u okviru jedne HTML strane moguće je na **proizvoljan način kombinovati** JavaScript i HTML kod
- **iz JavaScript-a moguće je generisati sam HTML kod**, u zavisnosti od akcije korisnika

HTML bez JavaScript-a

C:\aLutovac\2019\IP 2

File Edit Search View

primer01ax.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6
7
8              Primer
9              <br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>
10
11
12   </body>
13 </html>
```

Hyper Text Markup Language file

Primer 01a JavaScript

File | C:/aLutovac/2019/IP%202019/pr

Primer
Jedan
Dva
Tri

JavaScript u HTML

C:\aLutovac\2019\IP 2019\p

File Edit Search View Enc

primer01a.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer");
9          document.write("<br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file

Primer 01a JavaScript

File | C:/aLutovac/2019/IP%202019/pr

document.write("Primer");

Primer
Jedan
Dva
Tri

JavaScript u HTML

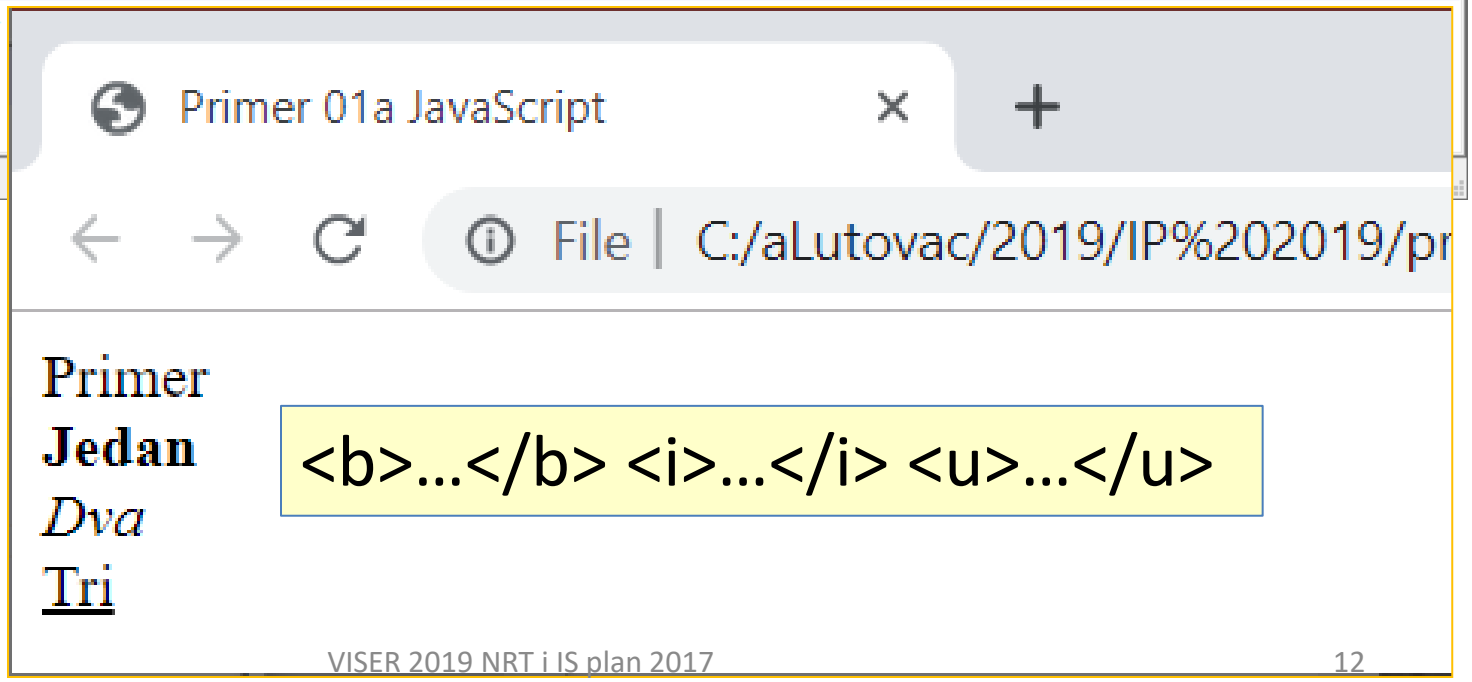
C:\aLutovac\2019\IP 2019\p

File Edit Search View Enc

primer01a.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer");
9          document.write("<br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file



JavaScript u HTML

C:\aLutovac\2019\IP 2019\

File Edit Search View Enc

primer01am1.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer");
9          document.write("<br><b>Jedan<br><i>Dva</i><br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file

Primer 01a JavaScript

File | C:/aLutovac/2019/IP%202019/pr

Primer
Jedan
Dva
Tri

nema

JavaScript u HTML

C:\aLutovac\2019\IP 20

File Edit Search View

primer01am2.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer");
9          document.write("<br><b>Jedan<br><i>Dva<br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file

Primer 01a JavaScript

File | C:/aLutovac/2019/IP%202019/pr

Primer
Jedan
Dva
Tri

nema i </i>

JavaScript u HTML

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer01am3.html - Notepad++

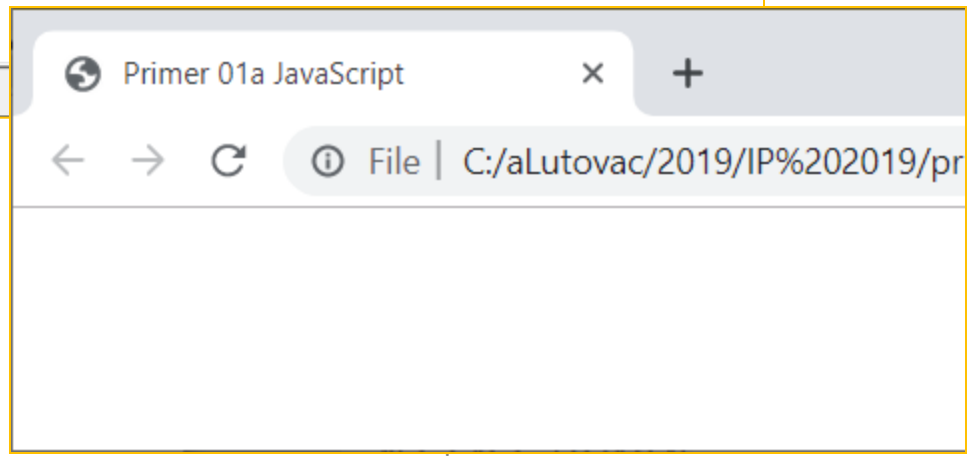
File Edit Search View

primer01am3.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer");
9          document.write("<br><b>Jedan
10         <br><i>Dva
11         <br><u>Tri</u><br>");
12       -->
13     </script>
14   </body>
15 </html>
```

Hyper Text Markup Language file

JavaScript u više redova **ne radi**!



JavaScript ↔ HTML

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer01am4.html - Notepad++

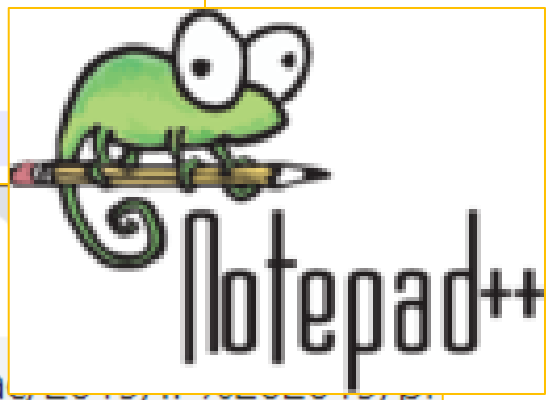
File Edit Search Vie

primer01am4.html

```
1  <html>
2    <head>
3      <title>Primer 01a JavaScript</title>
4    </head>
5    <body>
6
7
8              Primer
9              <br><b>Jedan
10             <br><i>Dva
11             <br><u>Tri</u><br>
12
13
14   </body>
15 </html>
```

Hyper Text Markup Language file

HTML u više redova **radi**

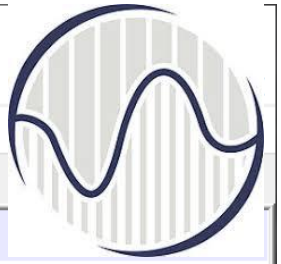


Primer 01a JavaScript

File | C:/aLutova

Primer
Jedan
Dva
Tri

JavaScript - poziv js fajla



C:\aLutovac\2019\IP 2019

File Edit Search View En

primer01spoljasnjiJS.html

```
1  <html>
2    <head>
3      <title>Primer 01 JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript" src="primer01spoljasnjiJS.js">
7      </script>
8    </body>
9  </html>
```

Hyper Text Markup Language file | length : 179 | lines : 9 | Ln : 1 | Col : 1

Primer 01 JavaScript

File | C:/aLutovac/2

Primer
Jedan
Dva
Tri

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer01spoljasnjiJS

File Edit Search View Encoding Language Settings Tools Macro R

primer01spoljasnjiJS.js

```
1  document.write("Primer");
2  document.write(
   "<br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>");
```

JavaScript file | length : 98 | lines : 2 | Ln : 1 | Col : 1 | Sel : 0 | 0 | Windows (CR LF) | UTF-8

JavaScript u HTML

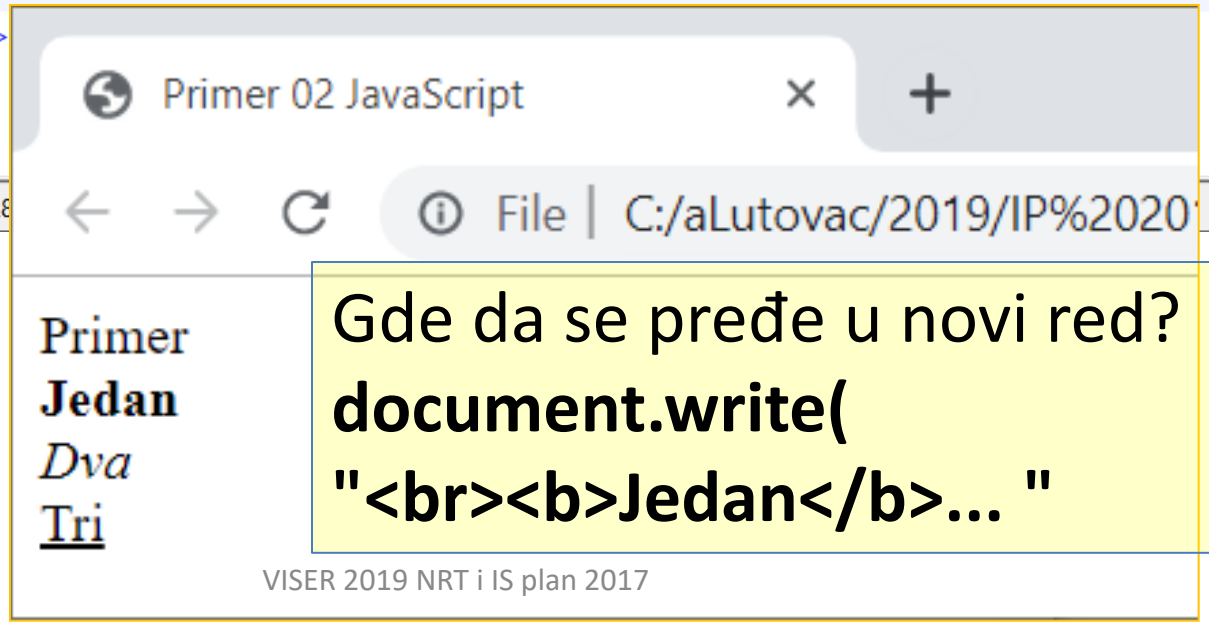
C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer02a.html - Notepad++

File Edit Search View

primer02a.html

```
1  <html>
2    <head>
3      <title>Primer 02 JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer")
9          document.write(
           "<br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file | length : 28 | INS



Gde da se pređe u novi red?

document.write(

**"
Jedan... "**

JavaScript u HTML

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer02a.html - Notepad++

File Edit Search View Er

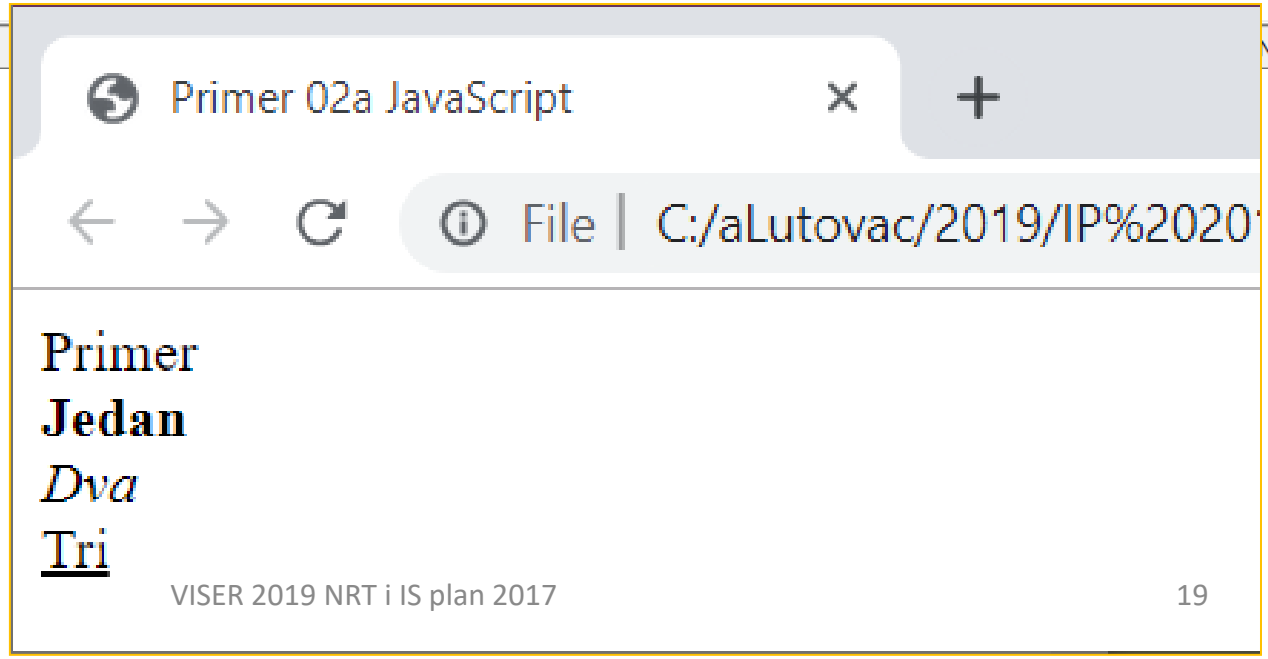
primer02b.html primer

```
1  <html>
2    <head>
3      <title>Primer 02a JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer")
9          document.write("<br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file

Poželjno ali ne mora

na kraju svake naredbe pisati simbol “;”



JavaScript u HTML

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer02b.html - Notepad++

File Edit Search View Er

primer02b.html

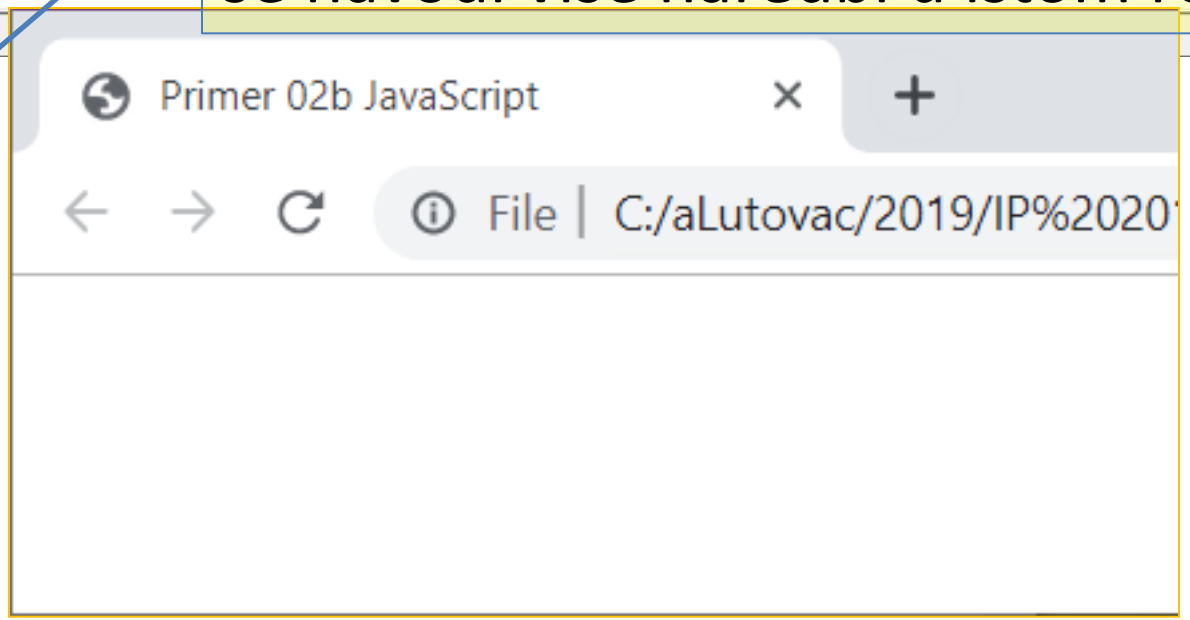
```
1  <html>
2    <head>
3      <title>Primer 02b JavaScript</title>
4    </head>
5    <body>
6      <script language="JavaScript">
7        <!--
8          document.write("Primer") document.write("<br>i dalje Primer")
9          document.write("<br><b>Jedan</b><br><i>Dva</i><br><u>Tri</u><br>");
10       -->
11     </script>
12   </body>
13 </html>
```

Hyper Text Markup Language file

INS

mora se koristiti tačka-zarez ako se navodi više naredbi u istom redu

ne radi



komentar

```
// komentar u jednoj liniji ...


/*
komentar u više redova ...
*/
```

Prikaz teksta

document.write(“neki tekst Prvi
”)

Imena promenljivih

- prvi znak mora da bude
  slovo engleske abecede ili simbol “_”
- mogu da sadrže brojeve i slova engleske abecede
- case sensitive jezik
- ne mogu se koristiti za imena promenljivih reči for, if, else, class, int...

Tipovi podataka

- celobrojni brojevi
- racionalni brojevi
- stringovi
- logički tip

Brojevi sa osnovom 10, 8 i 16

```
<script language="JavaScript">
  <!--
    c=0;
    a=1;
    b=255;
    c=a+b;
    document.write("a = ",a,",  b = ",b,",  c = a+b = ",c,"<br>");
    a=1;
    b='377';
    bOctal=parseInt(b,8);
    c=a+bOctal;
    document.write("a = ",a,",  b-octal = ",b,",  c = a+b = ",c,"<br>");
    a=1;
    b='7F';
    bHex=parseInt(b,16);
    c=a+bHex;
    document.write("a = ",a,",  b-hex = ",b,",  c = a+b = ",c,"<br>");
    a=1;
    b=255;
    bHex=b.toString(16);
    c=a+b;
    document.write("a = ",a,",  b-hex = ",bHex,",  c = a+b = ", a+b);
  -->
</script>
```

dekadni+dekadni

dekadni+oktalni

dekadni+heksa, prikaz string

heksa

a = 1, b = 255, c = a+b = 256
a = 1, b-octal = 377, c = a+b = 256
a = 1, b-hex = 7F, c = a+b = 128
a = 1, b-hex = ff, c = a+b = 256

Brojevi sa osnovom 10, 8 i 16

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer03b.html - Notepad++

File Edit Searc

primer03b.html

```
6   <script language="JavaScript">
7     <!--
8       b='0716';
9       document.write(
10      "b za osnovu 8 = ",b,", <br>",
11      "  b za osnovu 10 = ", parseInt(b,8),"<br><br>");
12      b='0x716';
13      document.write(
14      "b za osnovu 16 = ",b,", <br>",
15      "  b za osnovu 10 = ", parseInt(b,16),"<br>");
16    -->
17  </script>
```

Hyper Text Markup La length : 443 lines : 19 INS

parseInt

Primer 03b JavaScript

File | C:/aLutovac/2019/IP%202(

b za osnovu 8 = 0716,
b za osnovu 10 = 462

b za osnovu 16 = 0x716,
b za osnovu 10 = 1814

Racionalni brojevi

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer03c.html - Notepad++

File Edit Sear

primer03c.html

```
6   <script language="JavaScript">
7     <!--
8       b=3.14;
9       document.write("b = ",b,", <br>",
10      "  b  = ", b.toExponential(2),"<br><br>");
11      b=314E-2;
12      document.write("b = ",b,", <br>",
13      "  b  = ", b.toExponential(1),"<br><br>");
14      b=314e-2;
15      document.write("b = ",b,", <br>","  b  = ",
16      b.toExponential(4),"<br><br>");
17    -->
18  </script>
```

Hype length : 504 lines : 20 | Ln : 16 Col : 1 Sel : 0 | 0 | Windows (CR LF)

b = 3.14,
b = 3.14e+0

b = 3.14,
b = 3.1e+0

b = 3.14,
b = 3.1400e+0

.toExponential

Specijalni karakteri

****b = jedno mesto levo (*backspace*)

****f = jedan red nadole (*form feed*)

****n = početak novog reda (*new line character*)

****r = return (*carriage return*)

****t = tabulator (*tab*)

The 5 escape characters were originally designed to control typewriters, teletypes, and fax machines. They do not make any sense in HTML!

Racionalni brojevi

C:\aLutovac\2019\IP 2019\primeri JS\primeri prvi deo\primer03d.html - Notepad++

File Edit Sea

primer03d.html

```
6   <script language="JavaScript">
7     <!--
8       a="Prva rec";
9       b=" Druga\nrec";
10      document.write(a+b,"<br><br>");
11      b=" Druga rec";
12      document.write(a+'\n'+b,"<br><br>");
13      b=" Druga rec";
14      document.write(a+b,"<br><br>");
15    -->
16  </script>
```

length : 408 lines : 18 Ln : 14 Col : 41 Sel : 0 | 0 INS

Prva rec Druga rec

Prva rec Druga rec

Prva rec Druga rec

.toExponential

Logički tip podataka

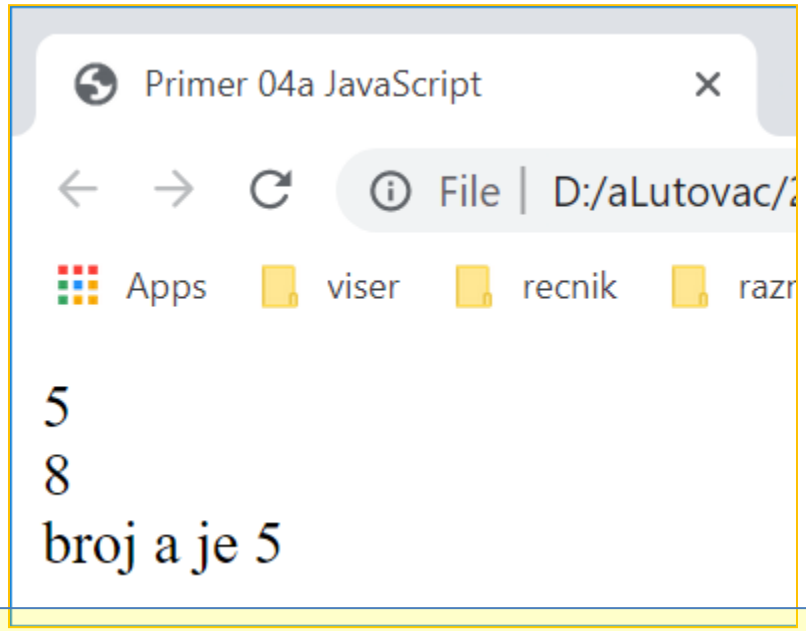
- dve vrednosti
  - true (tačno)
  - false (netačno)
- konverzija logičke vrednosti
  - true u broj 1
  - false u broj 0

Automatsko izvršavanje promene tipa

- dozvoljava da promenljiva ima različite tipove podataka u različito vreme izvršavanja

```
<script language="JavaScript">
  <!--
      a = 5;
      document.write(a,"<br>");
      b = 8;
      document.write(b,"<br>");
      b = "broj a je " + a;
      document.write(b,"<br>");
   -->
</script>
```

konkatenacija



broj 5 se konvertuje u tip String, "...5"

Aritmetički operatori

+ sabiranje,	+= sabiranje i dodela
- oduzimanje ,	-= oduzimanje i dodela
* množenje,	*= množenje i dodela
/ deljenje,	/= deljenje i dodela
% moduo,	%= moduo i dodela
++ inkrement,	-- dekrement

za matematičke operacije

Specijalna vrednost

- NaN (Not A Number) – ne zna se šta je
- Ukoliko je jedan od operanada tipa String za sve operatore, osim za sabiranje, pokušaće se da se izvede konverzija Stringa u broj i da se zatim izvrši operacija
- Kod sabiranja podatak koji nije tipa String konvertuje se u String i izvršava se sabiranje dva Stringa (konkatenacija)

Ostale operacije

- Unarni minus negira operand kojem prethodi
- Operator moduo kao rezultat vraća ostatak pri deljenju

Operatori na nivou bita

- Operacije nad celobrojnim brojevima dužine 32 bita
- Ukoliko operanad nije ceo broj dužine 32 bita, pokušaće se izvršiti konverzija u traženi tip, tek onda se primenjuje operacija
- Celobrojni brojevi se razmatraju na nivou bita i obavljaju operacije nad bitovima

Logičke operacije na nivou bita

- Logičko I (AND) a & b
  Rezultat se dobija 1, jedino ako su oba bita 1, u ostalim slučajevima rezultat je 0
- Logičko ILI (OR) a | b
  Rezultat se dobija 0, jedino ako su oba bita 0, u ostalim slučajevima rezultat je 1
- Logičko ekskluzivno ILI (XOR) a ^ b
  Rezultat se dobija 1, ako biti imaju različite vrednosti, u slučaju da imaju iste vrednosti, rezultat je 0
- Logičko NE (NOT) ~ a
  Komplementira bitove operanda a

Operacije pomeranja na nivou bita

- Pomeranje ulevo a << b
  Pomera binarni sadržaj operanda a za b mesta ulevo.
  Prazna mesta popunjava sa vrednošću 0
- Pomeranje udesno sa znakom a >> b
  Pomera binarni sadržaj operanda a za b mesta udesno.
  Prazna mesta popunjavju sa vrednošću najstarijeg bita
- Pomeranje udesno sa nulama a >>> b
  Pomera binarni sadržaj operanda a za b mesta udesno.
  Prazna mesta popunjavju sa vrednošću 0

Logički operatori

- I (&&) expr1 && expr2 Logičko I
- ILI (||) expr1 || expr2 Logičko ILI
- NE (!) !expr Negacija

Operatori poređenja

- Obavljaju poređenje dve vrednosti i kao rezultat vraćaju vrednost logičkog tipa
- Tip podataka celobrojan, racionalni, karakter, String i logički tip mogu se upoređivati koristeći operatore == i !=
- Samo numerički tipovi koriste ostale operatore

Operatori poređenja

- Jednakost (==)

  Rezultat je true, ako su operandi jednaki

  - vrši se konverzija podataka

- Nejednakost (!=)

  Rezultat je true, ako su operandi različiti

  - vrši se konverzija podataka

celobrojan, racionalni, karakter, string i logički tip

operatore vrednosti 5 i "5" su isti

Operatori poređenja

- Jednako bez konverzije tipova (===)
- Različito bez konverzije tipova (!==)

samo za brojeve

operatore vrednosti 5 i "5" su različiti

rezultat je false ili true

Operatori poređenja

- Veće (>)
  Rezultat je true, ako je levi operand veći od desnog operanda
- Veće ili jednako (>=)
  Rezultat je true, ako je levi operand veći ili jednak desnom

Samo za brojeve

- Manje (<)
  Rezultat je true, ako je levi operand manji od desnog operanda
- Manje ili jednako (<=)
  Rezultat je true, ako je levi operand manji ili jednak desnom
- Jednako bez konverzije tipova (===)
  Rezultat je true, ako su operandi jednaki bez konverzije
- Različito bez konverzije tipova (!==)
  Rezultat je true, ako su operandi različiti bez konverzije podataka

Predstavljanje brojeva

- brojevi su u dvostrukoj tačnosti sa pokretnom tačkom, IEEE 754 standard
- format je 64 bita
- za mantisu se koriste biti 0 do 51
- eksponent su biti 52 do 62
- znak je u bitu 63

Tačnost

- Celi brojevi do 15 cifara su tačni
- Najveći broj cifara je 17, ali aritmetika sa pokretnim zarezom nije uvek 100% tačna

```
<script language="JavaScript">
  <!--
     var a = 3.14;     // A number with decimals
     var b = 3;        // A number without decimals
    document.write("a = 3.14 =", a,"<br>","b = 3 = ", b,"<br><br>");
     var a = 123e5;     // 12300000
     var b = 123e-5;    // 0.00123
    document.write("a = 123e5 = ", a,"<br>","b = 123e-5 = ", b,"<br><br>");
     var a = 999999999999999;   // will be 999999999999999
     var b = 9999999999999999;  // will be 10000000000000000
    document.write("a = 999999999999999 = ", a,"<br>","b = 9999999999999999 = ", b,"<br><br>");
    var a = 0.2 + 0.1;         // will be 0.30000000000000004
    var b = 2/10 + 1/10;       // will be 0.30000000000000004
    document.write("a = 0.2 + 0.1 = ", a,"<br>","b = 2/10 + 1/10 = ", b,"<br><br>");
    var a = (0.2 * 10 + 0.1 * 10) / 10;       // will be 0.30000000000000004
    var b = (2 + 1)/10;        // will be 0.3
    document.write("a = (0.2 * 10 + 0.1 * 10) / 10 = ", a,"<br>","b = (2 + 1)/10 = ", b,"<br><br>");
  -->
</script>
```

a = 3.14 =3.14
b = 3 = 3

a = 123e5 = 12300000
b = 123e-5 = 0.00123

a = 999999999999999 = 999999999999999
b = 9999999999999999 = 10000000000000000

a = 0.2 + 0.1 = 0.30000000000000004
b = 2/10 + 1/10 = 0.30000000000000004

a = (0.2 * 10 + 0.1 * 10) / 10 = 0.3
b = (2 + 1)/10 = 0.3

Sabiranje brojeva i stringova

- Brojevi se sabiraju a stringovi nadovezuju

```
var x = 10;
var y = 20;
var z = x + y;          // will be 30 (a number)

var x = "10";
var y = "20";
var z = x + y;          // will be 1020 (a string)
```

Sabiranje brojeva i stringova

- Bojevi i stringovi

  brojevi se konvertuju u stringove

```
var x = "10";
var y = 20;
var z = x + y;          // will be 1020 (a string)
```

```
var x = 10;
var y = 20;
var z = "The result is: " + x + y;     //NIJE 30
```

Sabiranje brojeva i stringova

- Kompajler radi s leva u desno;
  prvo saber 10 + 20 i doda kao string sledećem:
  dobija se 30 + "30" zato što je naredni string

```
var x = 10;
var y = 20;
var z = "30";
var result = x + y + z;    //NIJE 102030
```

```
var a = 10;
var b = 20;
var c = a + b;              // will be 30 (a number)
document.write("a = 10 =", a,"<br>","b = 20 = ", b,"<br>");
document.write("c = ", c,"<br><br>");
```

a = 10 =10
b = 20 = 20
c = 30

```
var a = "10";;
var b = "20";;
var c = a + b;              // will be 1020 (a string)
document.write("a (= '10') =", a,"<br>","b (= '20') = ", b,"<br>");
document.write("c = ", c,"<br><br>");
```

a (= '10') =10
b (= '20') = 20
c = 1020

```
var a = 10;
var b = 20;
var c = "The result is not 30 =" + a + b;                    //NIJE 30
document.write("a = 10 =", a,"<br>","b = 20 = ", b,"<br>");
document.write("c = ", c,"<br><br>");
```

a = 10 =10
b = 20 = 20
c = The result is not 30 =1020

```
var a = 10;
var b = 20;
var c = "30";
var d = a + b + c;
document.write("a = 10 =", a,"<br>","b = 20 =", b,
"<br>","c (= '30') = ", c,"<br>");
document.write("d = ", d,"<br><br>");
```

a = 10 =10
b = 20 =20
c (= '30') = 30
d = 3030

```
var a = 10;
var b = 20;
var c = "30";
var d = a + b + c;
document.write("a = 10 =", a,"<br>","b = 20 =", b,
"<br>","c (= '30') = ", c,"<br>");
document.write("d = ", d,"<br><br>");

var a = 10;
var b = 20;
var c = "30";
var d = a + (b + c);
document.write("a = 10 =", a,"<br>","b = 20 =", b,
"<br>","c (= '30') = ", c,"<br>");
document.write("d = ", d,"<br><br>");
```

a = 10 =10
b = 20 =20
c (= '30') = 30
d = 3030

a = 10 =10
b = 20 =20
c (= '30') = 30
d = 102030

Sabiranje brojeva i stringova

- Brojevi se mogu zadati na dva načina

```
var x = "100";
var y = "10";
var z = x * y;       // will be 1000

var a = "100";
var b = "10";
var c = a - b;       // will be 90
```

```
var a ="100";
var b = "10";
var c = a * b;
document.write("a (= '100') = ", a,"<br>","b (= '10') =  ", b,"<br>");
document.write("c = ", c,"<br><br>");

var a = "100";
var b = "10";
var c = a - b;
document.write("a (= '100') =", a,"<br>","b (= '10') = ", b,"<br>");
document.write("c = ", c,"<br><br>");

var a = "100";
var b = "5";
var c = a / b;
document.write("a (= '100') =", a,"<br>","b (= '5') = ", b,"<br>");
document.write("c = ", c,"<br><br>");
```

a (= '100') = 100
b (= '10') = 10
c = 1000

a (= '100') =100
b (= '10') = 10
c = 90

a (= '100') =100
b (= '5') = 5
c = 20

Sabiranje brojeva i stringova

- Brojevi se mogu zadati na dva načina

```
var x = "100";
var y = "10";
var z = x + y;       // will be 10010, not 110
```

Samo kod sabiranja je konkatenacija

NaN - Not a Number

```
var x = 100 / "A";        // will be NaN

var x = 100 / "10";       // will be 10

var x = 100 / "Apple";
isNaN(x);          // returns true, it is Not a Number
```

Automatska konverzija brojeva

```
var a = 100;
var b = "10";
var c = a / b;
document.write("a (= 100) = ", a,"<br>","b (= '10') =  ", b,
document.write("c = ", c,"<br><br>");

var a = 100;
var b = "a";
var c = a / b;
document.write("a (= 100) =", a,"<br>","b (= 'a') = ", b,"<br>");
document.write("c = ", c,"<br>");
document.write("isNaN(c) = ", isNaN(c),"<br><br>");

var a = "100";
var b = a;
var c = a / b;
document.write("a (= '100') =", a,"<br>","b (= a) = ", b,"<br>");
document.write("c = ", c,"<br>");
document.write("isNaN(c) = ", isNaN(c),"<br><br>");
```

a (= 100) = 100
b (= '10') = 10
c = 10

a (= 100) =100
b (= 'a') = a
c = NaN
isNaN(c) = true

a (= '100') =100
b (= a) = 100
c = 1
isNaN(c) = false

```
var a = NaN;
var b = 10;
var c = a + b;
document.write("a (= NaN) = ", a,"<br>","b (= 10) =  ", b,"<br>");
document.write("typeof c = ", typeof c,"<br>");
document.write("c = ", c,"<br><br>");

var a = NaN;
var b = "10";
var c = a + b;
document.write("a (= NaN) = ", a,"<br>","b (= '10') =  ", b,"<br>");
document.write("c = ", c,"<br>");
document.write("typeof c = ", typeof c,"<br><br>"

document.write("isNaN(a) = ", isNaN(a),"<br>");
document.write("typeof a = ", typeof a,"<br>");
document.write("typeof b = ", typeof b,"<br>");
document.write("typeof c = ", typeof c,"<br><br>"
```

a (= NaN) = NaN
b (= 10) = 10
typeof c = number
c = NaN

a (= NaN) = NaN
b (= '10') = 10
c = NaN10
typeof c = string

isNaN(a) = true
typeof a = number
typeof b = string
typeof c = string

NaN - Not a Number

```
typeof NaN;          // returns "number"
```

NaN - Not a Number

```
var x = NaN;
var y = 5;
var z = x + y;        // will be NaN
```

Svaka operacija sa brojevima gde je jedan činilac broj NaN, daje NaN

type of x daje number

```
var x = NaN;
var y = "5";
var z = x + y;        // will be NaN5
```

konkatenacija

Infinity

- Infinity (or -Infinity) se dobija kada je broj izvan opsega

```
var x =  2 / 0;          // will be Infinity
var y = -2 / 0;          // will be -Infinity

typeof Infinity;         // returns "number"
```

```
var a = 10;
var b = 0;
var c = a / b;
document.write("a  = ", a,"<br>","b  =  ", b,"<br>");
document.write("typeof c = ", typeof c,"<br>");
document.write("c = ", c,"<br><br>");
var a = -10;
var b = 0;
var c = a / b;
document.write("a  = ", a,"<br>","b  =  ", b,"<br>");
document.write("c = ", c,"<br><br>");
var a = 0;
var b = 0;
var c = a / b;
document.write("a  = ", a,"<br>","b  =  ", b,"<br>");
document.write("c = ", c,"<br><br>");
var a = 10/0;
var b = -10/0;
var c = a + b;
document.write("a  = ", a,"<br>","b  =  ", b,"<br>");
document.write("c = ", c,"<br><br>");
```

a = 10
b = 0
typeof c = number
c = Infinity

a = -10
b = 0
c = -Infinity

a = 0
b = 0
c = NaN

a = Infinity
b = -Infinity
c = NaN

```
var a = 1;
var b = 0.2;
c=a-b; c=c-b;c=c-b;c=c-b;c=c-b;
document.write("a  = ", a,"<br>","b  =  ", b,"<br>");
document.write("typeof c = ", typeof c,"<br>");
document.write("c = ", c,"<br>");
document.write(c==0,"<br><br>");

var a = 10;
var b = 2;
c=a-b; c=c-b;c=c-b;c=c-b;c=c-b;
document.write("a  = ", a,"<br>","b  =  ", b,"<br>");
document.write("typeof c = ", typeof c,"<br>");
document.write("c = ", c,"<br>");
document.write(c==0,"<br><br>");
```

a = 1
b = 0.2
typeof c = number
c = 5.551115123125783e-17
false

a = 10
b = 2
typeof c = number
c = 0
true

Osnova 16, 8, 2

```
var x = 0xFF;          // will be 255

var myNumber = 128;
myNumber.toString(16);  // returns 80
myNumber.toString(8);   // returns 200
myNumber.toString(2);   // returns 10000000
```

Osnova bojeva je 10 decimals.
toString() method se koristi da se dobije broj u osnovi 16 (hex), 8 (octal) ili 2 (binary)

Brojevi mogu biti objekti

```
var x = 123;              // typeof x returns number
var y = new Number(123);
                          // typeof y returns object
var x = 500;
var y = new Number(500);
  // (x == y) is true because x and y have equal values
```

Nemojte da pravite Number objects.
Usporava brzinu izvršavanja.
Ključna reč **new** komplikuje kod.
Može da uzrokuje neočekivane rezultate!

Brojevi mogu biti objekti

```
var x = 500;

var y = new Number(500);

// (x === y) is false because x and y have different types


var x = new Number(500);
var y = new Number(500);
// (x == y) is false because objects cannot be compared
```

Objekti ne mogu da se upoređuju!
Upoređivanje dva JavaScript objekta vraća false.

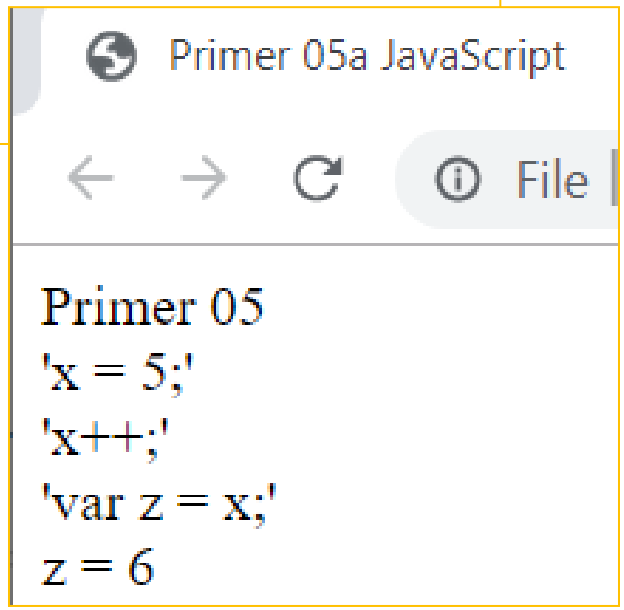
```
var a = 123;
var b = new Number(123);
document.write("a  = ", a,"<br>");
document.write("b  = ", b,"<br>");
document.write("typeof a = ", typeof a,"<br>");
document.write("typeof b = ", typeof b,"<br>");
document.write("a == b = ", (a == b),"<br>");
document.write("a === b = ",
(a === b),"<br><br>");

var a = new Number(123);
var b = new Number(123);
document.write("a  = ", a,"<br>");
document.write("b  = ", b,"<br>");
document.write("typeof a = ", typeof a,"<br>");
document.write("typeof b = ", typeof b,"<br>");
document.write("a == b = ", (a == b),"<br>");
document.write("a === b = ",
(a === b),"<br><br>");
```

a = 123
b = 123
typeof a = number
typeof b = object
a == b = true
a === b = false

a = 123
b = 123
typeof a = object
typeof b = object
a == b = false
a === b = false

```html
<html>
  <head>
    <title>Primer 05a JavaScript</title>
  </head>
  <body>
    <script language="JavaScript" src="primer05a.js">
    </script>
  </body>
</html>
```

primer05a.html




```javascript
var x = 5;
x++;
var z = x;
document.write("Primer 05");
document.write("<br>'x = 5;' <br>'x++;'<br>'var z = x;' <br>z = ",z);
```

primer05a.js

primer05b.html

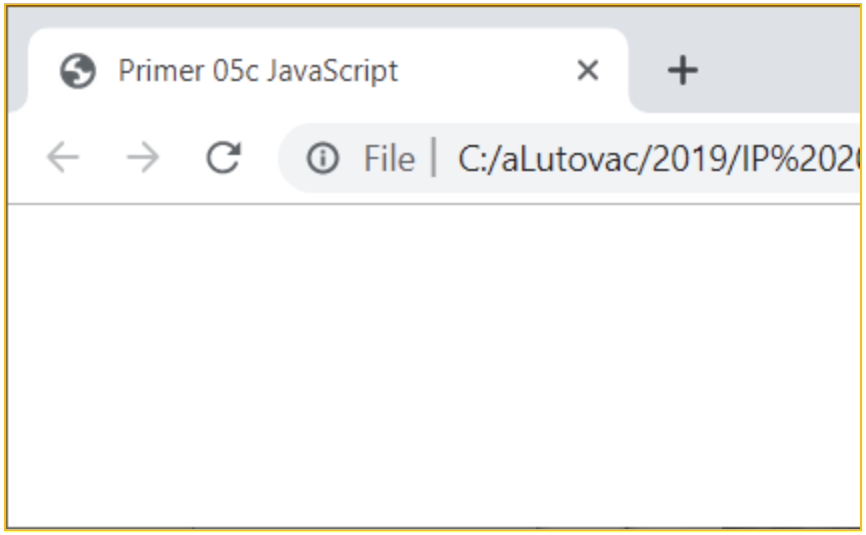
```html
<html>
  <head>
    <title>Primer 05b JavaScript</title>
  </head>
  <body>
    <script language="JavaScript" src="primer05b.js">
    </script>
  </body>
</html>
```

```
txt1 = 'What a very ';
txt1 += ' nice day';'
txt2 = txt1;
txt2 = What a very nice day
```

primer05b.js

```javascript
txt1 = 'What a very ';
txt1 += '  nice day';
txt2 = txt1;
document.write("<br><br>txt1 = 'What a very ';");
document.write(
"<br>txt1 += '   nice day';'<br>txt2 = txt1;<br>txt2 = ",
 txt2);
```

txt1 ++= ' nice day';
greška!




```
document.write("txt1 = 'What a very ';");
document.write("<br>txt1 ++= '   nice day';'<br>txt2 = txt1; ");
txt1 = 'What a very ';
txt1 ++= '  nice day';
txt2 = txt1;
document.write("<br>txt2 = ", txt2);
```

primer05c.js

primer05d.js

primer05d.js

```
1  var x = 5 & 1;
2
3  document.write("Primer 05d.js");
4  document.write("<br>x = 5 & 1;");
5  document.write("<br>0101 & 0001 = 0001 ");
6  document.write("<br>x = ",x);
7
8  var a = 17;
9  document.write("<br><br>a = ",a);
10 document.write("<br>a = ",a.toString(2));
11 var b = 1;
12 document.write("<br>b = ",b.toString(2));
13 var c = a & b;
14 document.write("<br>c = ",c.toString(2));
```

Primer 05d.js
x = 5 & 1;
0101 & 0001 = 0001
x = 1

a = 17
a = 10001
b = 1
c = 1

primer05e.js

primer05e.js

```
1  var x = 5 | 2;
2  document.write("Primer 05e.js");
3  document.write("<br>x = 5 | 2;");
4  document.write("<br>0101 | 0010 = 0111 "
5  document.write("<br>x = ",x.toString(2));
6  document.write("<br>x = ",x);
7
8  var a = 18;
9  document.write("<br><br>a = ",a);
10 document.write("<br>a = ",a.toString(2));
11 var b = 1;
12 document.write("<br>b = ",b.toString(2));
13 var c = a | b;
14 document.write("<br>var c = a | b = ",c);
15 document.write("<br>c = ",c.toString(2));
```

Primer 05e.js
x = 5 | 2;
0101 | 0010 = 0111
x = 111
x = 7

a = 18
a = 10010
b = 1
var c = a | b = 19
c = 10011

Primer 05f JavaScript

File

Primer 05f
x = 5
~x = -6
x&(~x) = 0

x = 101
~x = -110
x&(~x) = 0

primer05f.js

primer05f.js

```
1  var x = 5;
2  var y = ~x;
3  document.write("Primer 05f");
4  document.write("<br>x = ",x);
5  document.write("<br>~x = ",y);
6  document.write("<br>x&(~x) = ",x&y);
7
8  document.write("<br><br>x = ",x.toString(2));
9  document.write("<br>~x = ",y.toString(2));
10 document.write("<br>x&(~x) = ",(x&y).toString(2));
```

```
a = 5
BaseForm[a, 2]
IntegerString[a, 2]
IntegerString[a, 2] // InputForm
```

5

101_2

101

"101"

```
b = BitNot[a]
BaseForm[b, 2]
IntegerString[b, 2]
IntegerString[b, 2] // InputForm
```

−6

-110_2

110

"110"

Primer 05f

x = 5

~x = -6

x&(~x) = 0

x = 101

~x = -110

x&(~x) = 0

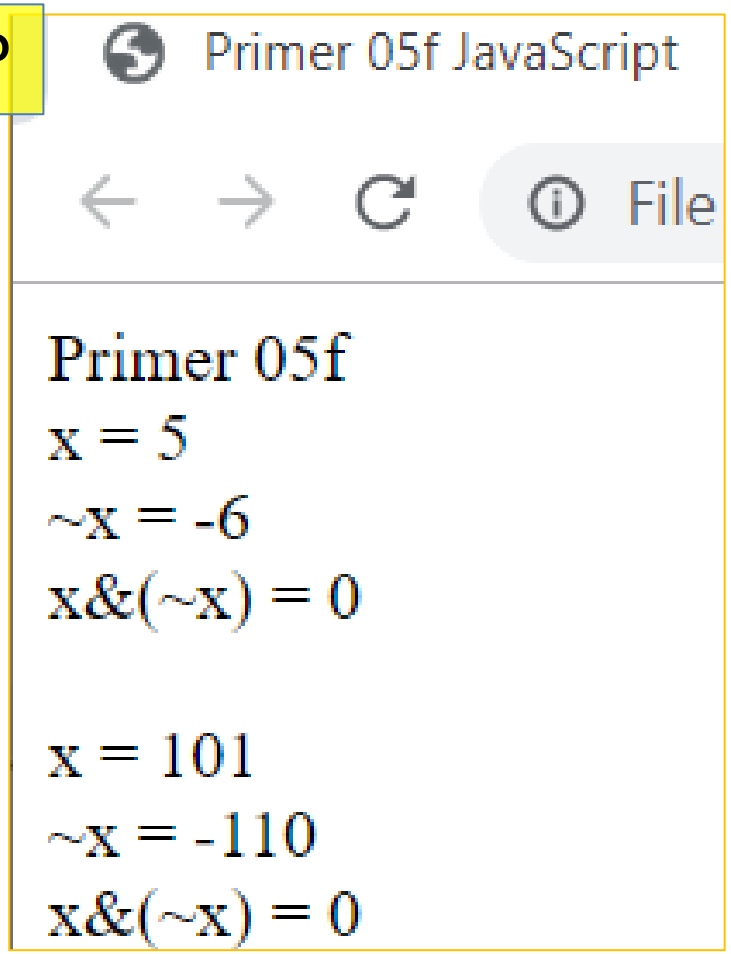
```
BitNot[0]
```

−1

Primer sa 4 bita

0101 & 0001	0001
0101 | 0001	0101
~0101	1010
0101 << 1	1010
0101 ^ 0001	0100
0101 >> 1	0010
0101 >>> 1	0010

greška?



Primer sa 4 bita je unsigned binary numbers
Zato ~5 returns 10

00000000000000000000000000000101 (5)

11111111111111111111111111111010 (~5 = -6)

JavaScript koristi 32 bita signed integers
Zato ~5 daje -6

Primer sa 4 bita je unsigned binary numbers
Zato ~5 daje 10

nije greška!
signed integer koriste leftmost bit kao minus znak

Provera u drugom softveru

```
a = 5; b = 1;
{a, b}
{BitAnd[a, b], BitOr[a, b], BitNot[a],
 BitShiftLeft[a], BitShiftRight[a], BitXor[a, b]}
IntegerString[{
    BitAnd[a, b],
    BitOr[a, b],
    BitNot[a],
    BitShiftLeft[a],
    BitShiftRight[a],
    BitXor[a, b]
  }, 2] // InputForm
```

```
{5, 1}

{1, 5, -6, 10, 2, 4}

{"1", "101", "110", "1010", "10", "100"}
```

0101 & 0001	0001
0101 | 0001	0101
~0101	1010
0101 << 1	1010
0101 ^ 0001	0100
0101 >> 1	0010
0101 >>> 1	0010

5	00000000000000000000000000000101
1	00000000000000000000000000000001
5 & 1	00000000000000000000000000000001 (1)

JavaScript Bitwise AND (&)

5	00000000000000000000000000000101
1	00000000000000000000000000000001
5 | 1	00000000000000000000000000000101 (5)

JavaScript Bitwise OR (|)

JavaScript Bitwise XOR (^)

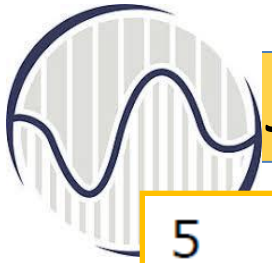
5	00000000000000000000000000000101
1	00000000000000000000000000000001
5 ^ 1	00000000000000000000000000000100 (4)

primer05g.js

```
1  var x = 5 ^ 1;
2  document.write("Primer 05g");
3  document.write("<br>x = 5 ^ 1;");
4  document.write("<br>0101 ^ 0001 = 0100 ");
5  document.write("<br>x = ",x.toString(2));
6  document.write("<br>x = ",x);
7
8  var a = 18;
9  document.write("<br><br>a = ",a);
10 document.write("<br>a = ",a.toString(2));
11 var b = 1;
12 document.write("<br>b = ",b.toString(2));
13 var c = a ^ b;
14 document.write("<br>var c = a ^ b = ",c);
15 document.write("<br>c = ",c.toString(2));
```

Primer 05g
x = 5 ^ 1;
0101 ^ 0001 = 0100
x = 100
x = 4

a = 18
a = 10010
b = 1
var c = a ^ b = 19
c = 10011



JavaScript (Zero Fill) Bitwise Left Shift (<<)

5	00000000000000000000000000000101
5 << 1	00000000000000000000000000001010 (10)

primer05h.js

```
1  var x = 5 << 1;
2  document.write("Primer 05h");
3  document.write("<br>x = 5 << 1;");
4  document.write("<br>0101 << 0001 = 1010 ");
5  document.write("<br>x = ",x.toString(2));
6  document.write("<br>x = ",x);
7
8  var a = 18;
9  document.write("<br><br>a = ",a);
10 document.write("<br>a = ",a.toString(2));
11 var b = 1;
12 document.write("<br>b = ",b.toString(2));
13 var c = a << b;
14 document.write("<br>var c = a << b = ",c);
15 document.write("<br>c = ",c.toString(2));
```

Primer 05h
x = 5 << 1;
0101 << 0001 = 1010
x = 1010
x = 10

a = 18
a = 10010
b = 1
var c = a << b = 36
c = 100100

JavaScript (Sign Preserving) Bitwise Right Shift (>>)

-5	11111111111111111111111111111011
-5 >> 1	11111111111111111111111111111101 (-3)

primer05i.js

```
1  var x = -5 >> 1;
2  document.write("Primer 05i");
3  document.write("<br>x = -5 >> 1;");
4  document.write("<br>x = ",x.toString(2));
5  document.write("<br>x = ",x);
6
7  var a = 18;
8  document.write("<br><br>a = ",a);
9  document.write("<br>a = ",a.toString(2));
10 var b = 1;
11 document.write("<br>b = ",b.toString(2));
12 var c = a >> b;
13 document.write("<br>var c = a >> b = ",c);
14 document.write("<br>c = ",c.toString(2));
```

Primer 05i
x = -5 >> 1;
x = -11
x = -3

a = 18
a = 10010
b = 1
var c = a >> b = 9
c = 1001

JavaScript (Zero Fill) Right Shift (>>>)

5	00000000000000000000000000000101
5 >>> 1	00000000000000000000000000000010 (2)

Primer 05j
x = 5 >>> 1;
x = 10
x = 2
x = -5 >>> 1;
x = 11111111111111111111111111111101
x = 2147483645

primer05j.js

```
1  var x = 5 >>> 1;
2  document.write("Primer 05j");
3  document.write("<br>x = 5 >>> 1;");
4  document.write("<br>x = ",x.toString(2));
5  document.write("<br>x = ",x);
6
7  var x = -5 >>> 1;
8  document.write("<br>x = -5 >>> 1;");
9  document.write("<br>x = ",x.toString(2));
10 document.write("<br>x = ",x);
```

Binary Representation	Decimal value
00000000000000000000000000000001	1
00000000000000000000000000000010	2
00000000000000000000000000000100	4
00000000000000000000000000001000	8
00000000000000000000000000010000	16
00000000000000000000000000100000	32
00000000000000000000000001000000	64

Binary Representation	Decimal value
00000000000000000000000000000101	5 (4 + 1)
00000000000000000000000000001101	13 (8 + 4 + 1)
00000000000000000000000000101101	45 (32 + 8 + 4 + 1)

00000000000000000000000000000101	5
11111111111111111111111111111011	-5
00000000000000000000000000000110	6
11111111111111111111111111111010	-6
00000000000000000000000000101000	40
11111111111111111111111111011000	-40

JavaScript binarni brojevi su u formatu komplementa 2
Negativni brojevi se dobijaju kao NOT svih bitova plus 1

00000000000000000000000000000101	5
11111111111111111111111111111011	-5
00000000000000000000000000000110	6
11111111111111111111111111111010	-6
00000000000000000000000000101000	40
11111111111111111111111111011000	-40

primer05k.js

```
1  var x = 5;
2  var y = (~x)+1;
3  document.write("Primer 05k");
4  document.write("<br>x = 5;");
5  document.write("<br>x = ",x);
6  document.write("<br>x = ",x.toString(2));
7  document.write("<br>y = (~x)+1;");
8  document.write("<br>y = ",y.toString(2));
9  document.write("<br>y = ",y);
10
11 var x = -6;
12 var y = (~x)+1;
13 document.write("<br><br>x = -6;");
14 document.write("<br>x = ",x);
15 document.write("<br>x = ",x.toString(2));
16 document.write("<br>y = (~x)+1;");
17 document.write("<br>y = ",y.toString(2));
18 document.write("<br>y = ",y);
```

Primer 05k
x = 5;
x = 5
x = 101
y = (~x)+1;
y = -101
y = -5

x = -6;
x = -6
x = -110
y = (~x)+1;
y = 110
y = 6

sa 3 bita

Bits	Unsigned value	Two's complement value
011	3	3
010	2	2
001	1	1
000	0	0
111	7	−1
110	6	−2
101	5	−3
100	4	−4

sa 8 bita

Bits	Unsigned value	Two's complement value
0111 1111	127	127
0111 1110	126	126
0000 0010	2	2
0000 0001	1	1
0000 0000	0	0
1111 1111	255	−1
1111 1110	254	−2
1000 0010	130	−126
1000 0001	129	−127
1000 0000	128	−128

JavaScript koristi
32-bit signed brojeve

JavaScript koristi 32-bit signed brojeve

primer06a.js

```
1  var x = 5;
2  var y = dec2bin(x);
3  var z = bin2dec(y);
4
5  document.write("<br>x = ",x);
6  document.write("<br>x_bin = ",x.toString(2));
7  document.write("<br>y = ",y);
8  document.write("<br>z = ",z);
9
10 function dec2bin(dec){
11     return (dec >>> 0).toString(2);
12 }
13
14 function bin2dec(bin){
15     return parseInt(bin, 2).toString(10);
16 }
```

x = 5
x_bin = 101
y = 101
z = 5

function

primer06b.js

```
1  var x = 5;
2  var y = dec2bin(x);
3  var z = bin2dec(y);
4
5  document.write("x = ",x);
6  document.write("<br>x_bin = ",x.toString(2));
7  document.write("<br>y = ",y);
8  document.write("<br>z = ",z);
9
10 document.write("<br><br>x + 2 = ",x+2);
11 document.write("<br>y + 2 = ",y+2);
12 document.write("<br>z + 2 = ",z+2);
13
14 document.write("<br><br>x je "+x);
15 document.write("<br>y je "+y);
16 document.write("<br>z je "+z);
17
18
19 function dec2bin(dec){
20     return (dec >>> 0).toString(2);
21 }
22
23 function bin2dec(bin){
24     return parseInt(bin, 2).toString(10);
25 }
```

x = 5
x_bin = 101
y = 101
z = 5

x + 2 = 7
y + 2 = 1012
z + 2 = 52

x je 5
y je 101
z je 5

Automatska konverzija brojeva

primer06c.js

```
1  var x = 5;
2  var y = 1*dec2bin(x);
3  var z = 1*bin2dec(y);
4
5  document.write("x = ",x);
6  document.write("<br>x_bin = ",x.toString(2));
7  document.write("<br>y = ",y);
8  document.write("<br>z = ",z);
9
10 document.write("<br><br>x + 2 = ",x+2);
11 document.write("<br>y + 2 = ",y+2);
12 document.write("<br>z + 2 = ",z+2);
13
14 document.write("<br><br>x je "+x);
15 document.write("<br>y je "+y);
16 document.write("<br>z je "+z);
17
18
19 function dec2bin(dec){
20     return (dec >>> 0).toString(2);
21 }
22
23 function bin2dec(bin){
24     return parseInt(bin, 2).toString(10);
25 }
```

```
x = 5
x_bin = 101
y = 101
z = 5

x + 2 = 7
y + 2 = 103
z + 2 = 7

x je 5
y je 101
z je 5
```

primer06c2.js

```
1  var x = 5;
2  var y = dec2bin(x);
3  var z = bin2dec(y);
4
5  document.write("x = ",x);
6  document.write("<br>x_bin = ",x.toString(2));
7  document.write("<br>y = ",y);
8  document.write("<br>z = ",z);
9
10 document.write("<br><br>x + 2 = ",x+2);
11 document.write("<br>y + 2 = ",y+2);
12 document.write("<br>z + 2 = ",z+2);
13
14 document.write("<br><br>x je "+x);
15 document.write("<br>y je "+y);
16 document.write("<br>z je "+z);
17
18
19 function dec2bin(dec){
20     return (dec >>> 0).toString(2);
21 }
22
23 function bin2dec(bin){
24     return parseInt(bin, 2).toString(10);
25 }
```

nema 1*dec2bin(x);
nema 1*bin2dec(y);

x = 5
x_bin = 101
y = 101
z = 5

x + 2 = 7
y + 2 = 1012
z + 2 = 52

x je 5
y je 101
z je 5

Automatska konverzija brojeva

primer06d.js

```
1  var x = 100 + 50;;
2  var y = 100 - 50;;
3  var z = "100" + "50";;
4  var v = "100" - "50";;
5  var w = "100 - 50";;
6
7  document.write("x = ",x);
8  document.write("<br>y = ",y);
9  document.write("<br>z = ",z);
10 document.write("<br>v = ",v);
11 document.write("<br>w = ",w);
```

x = 150
y = 50
z = 10050
v = 50
w = 100 - 50

Dvostruki klik na fajl sa ekstenzijom .js